*Fig. 1*

2/22

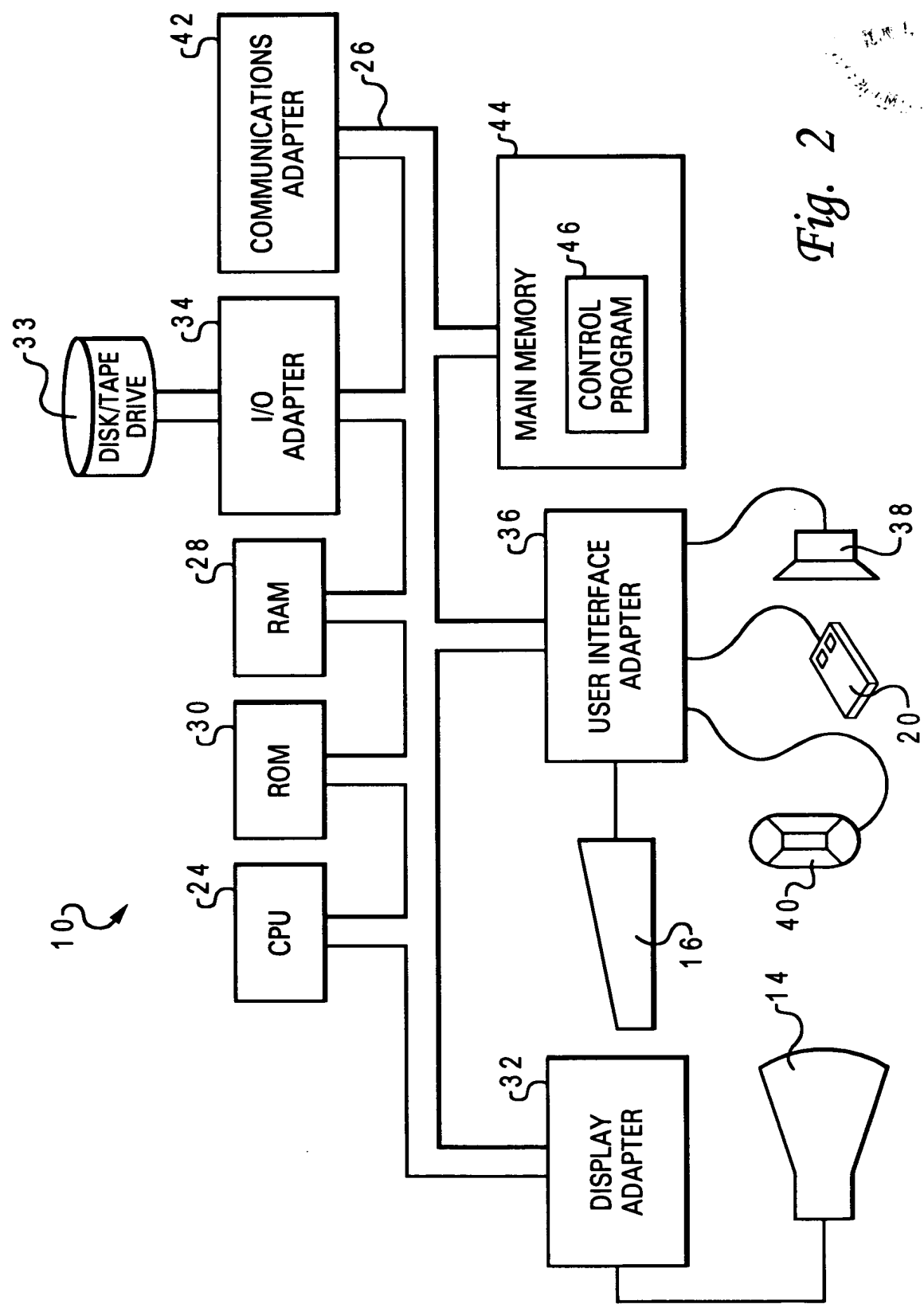
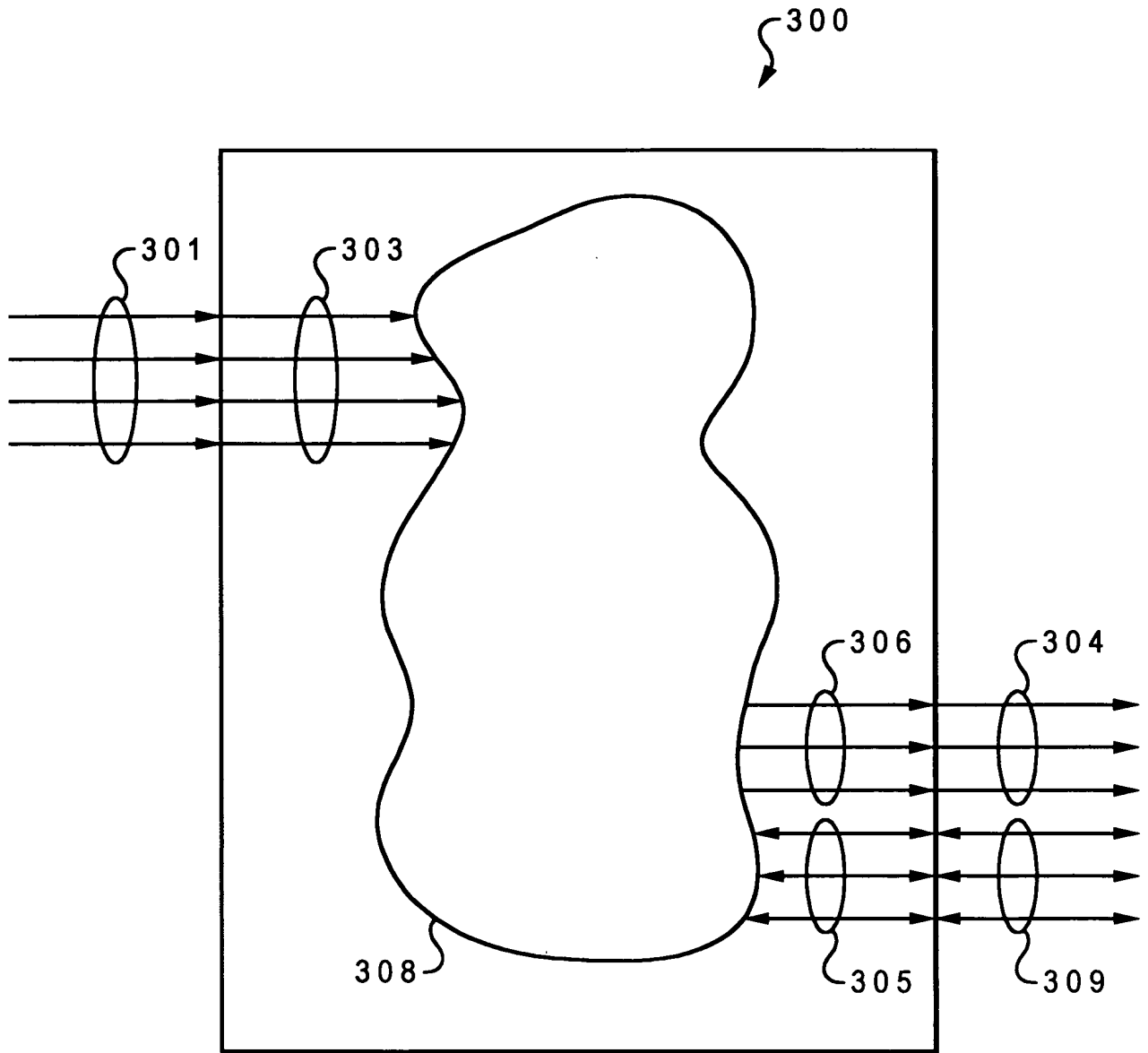


Fig. 2



*Fig. 3A*

4/22

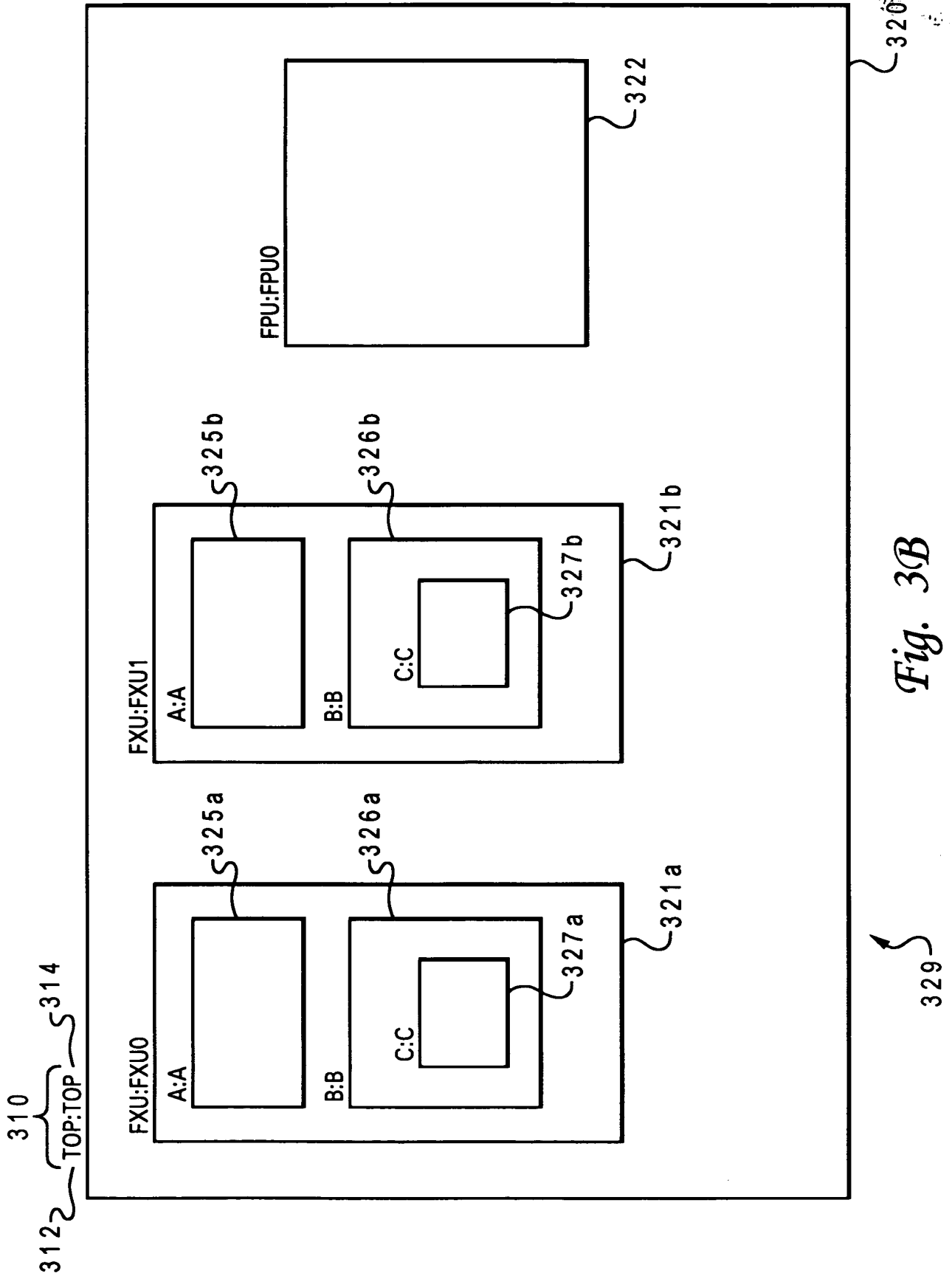
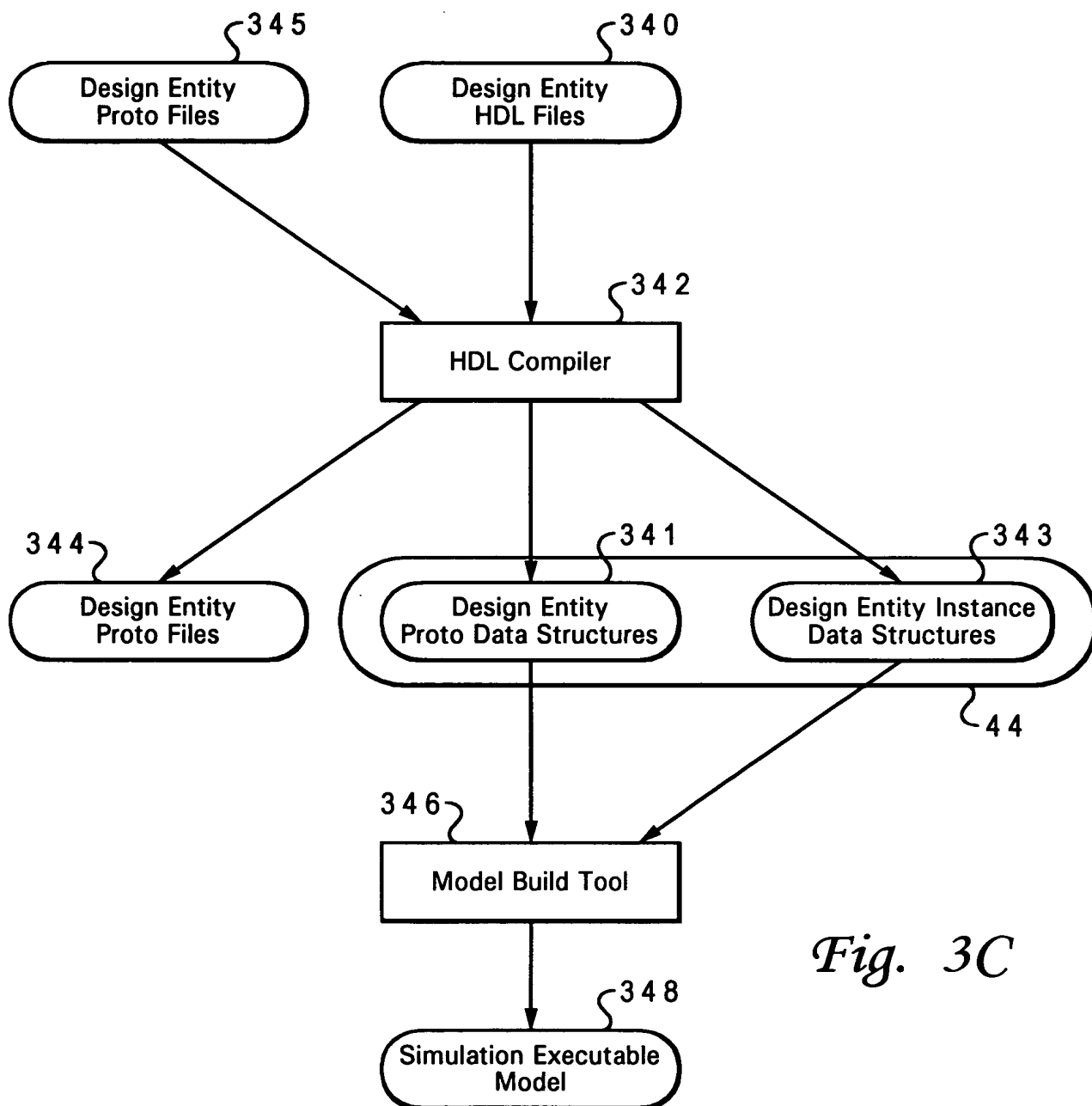


Fig. 3B

5/22

*Fig. 3C*

6/22

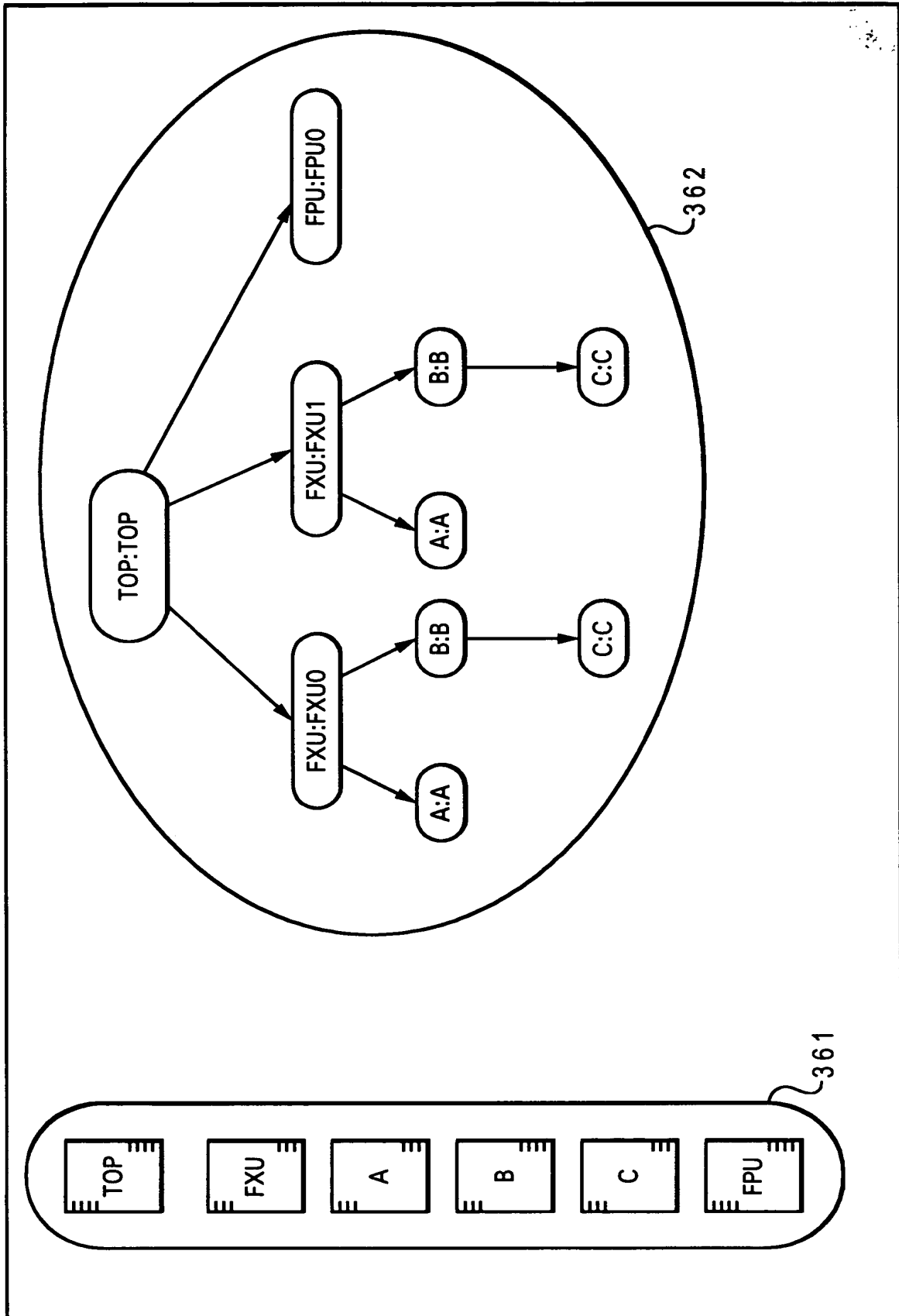


Fig. 3D

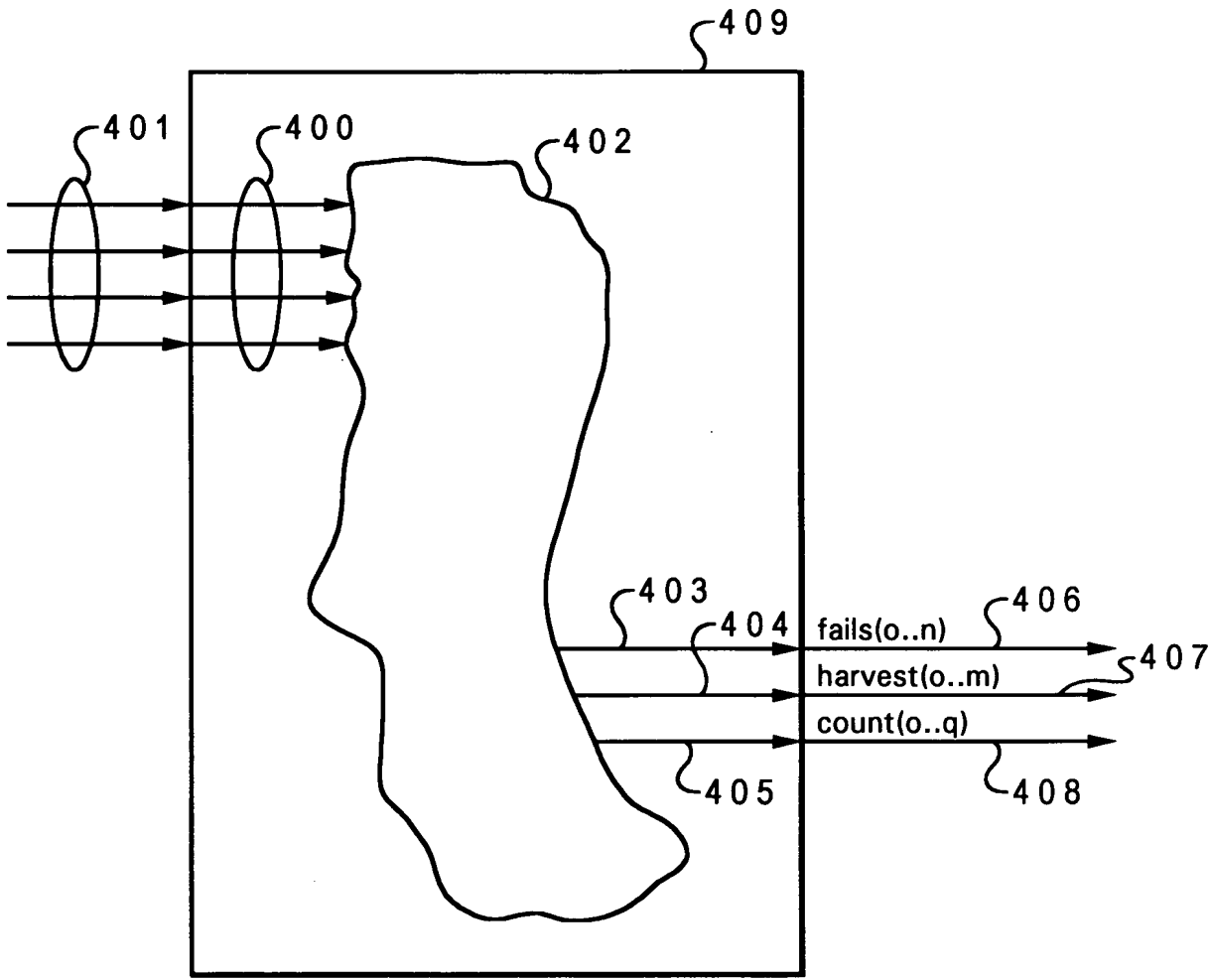
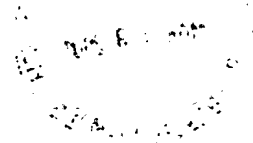
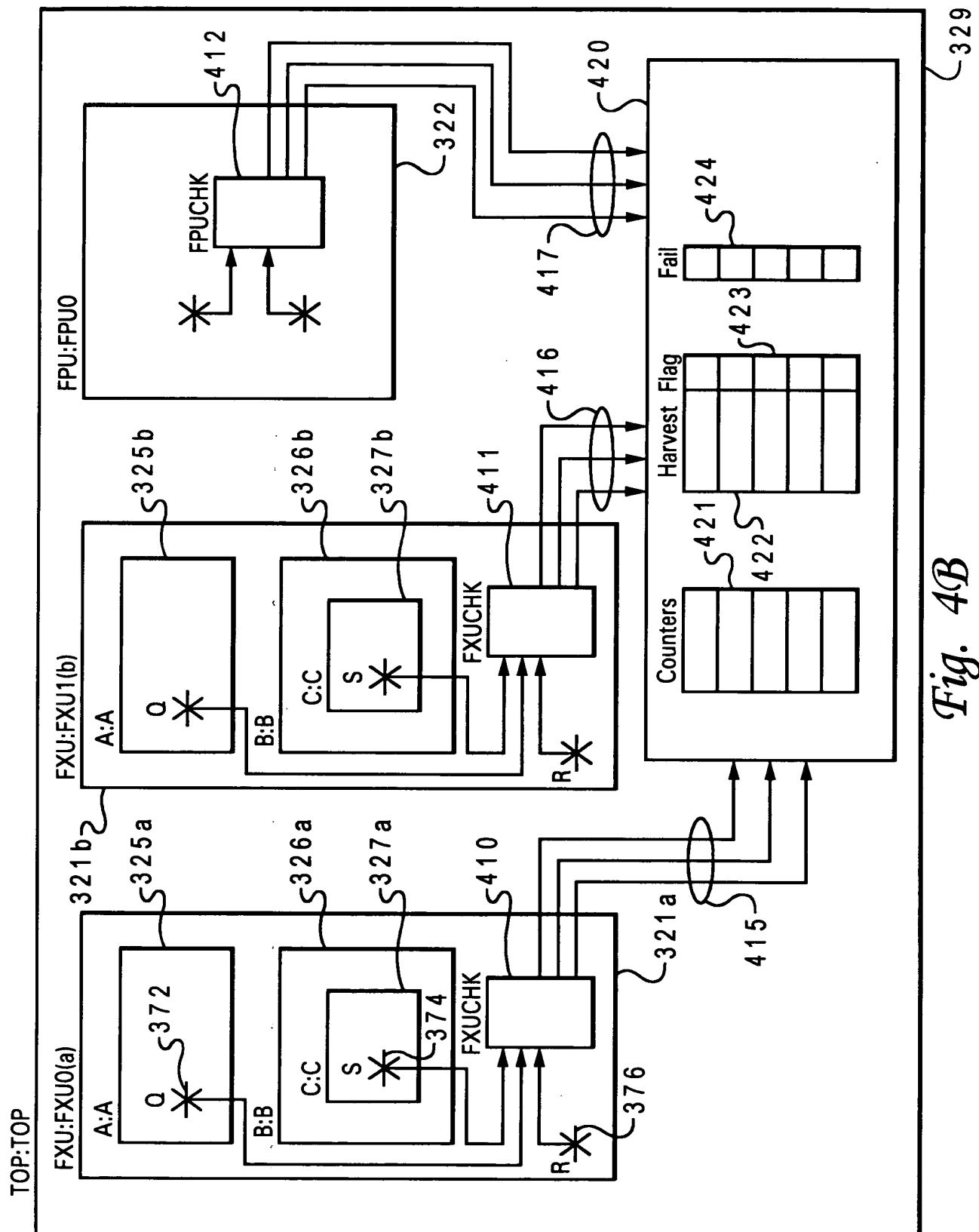


Fig. 4A





9/22

ENTITY FXUCHK IS

```

PORT(  S_IN      :  IN std_ulogic;
        Q_IN      :  IN std_ulogic;
        R_IN      :  IN std_ulogic;
        clock      :  IN std_ulogic;
        fails      :  OUT std_ulogic_vector(0 to 1);
        counts     :  OUT std_ulogic_vector(0 to 2);
        harvests   :  OUT std_ulogic_vector(0 to 1);
);

```

```

4 5 2 { --!! BEGIN
      --!! Design Entity: FXU;

```

```

4 5 3 { --!! Inputs
      --!! S_IN      =>    B.C.S;
      --!! Q_IN      =>    A.Q;
      --!! R_IN      =>    R;
      --!! CLOCK     =>    clock;
      --!! End Inputs

```

```

4 5 4 { --!! Fail Outputs;
      --!! 0 : "Fail message for failure event 0";
      --!! 1 : "Fail message for failure event 1";
      --!! End Fail Outputs;

```

```

4 5 5 { --!! Count Outputs;
      --!! 0 : <event0> clock;
      --!! 1 : <event1> clock;
      --!! 2 : <event2> clock;
      --!! End Count Outputs;

```

```

4 5 6 { --!! Harvest Outputs;
      --!! 0 : "Message for harvest event 0";
      --!! 1 : "Message for harvest event 1";
      --!! End Harvest Outputs;

```

```

4 5 7 { --!! End;

```

ARCHITECTURE example of FXUCHK IS

BEGIN

... HDL code for entity body section ...

END;

*Fig. 4C*

10/22

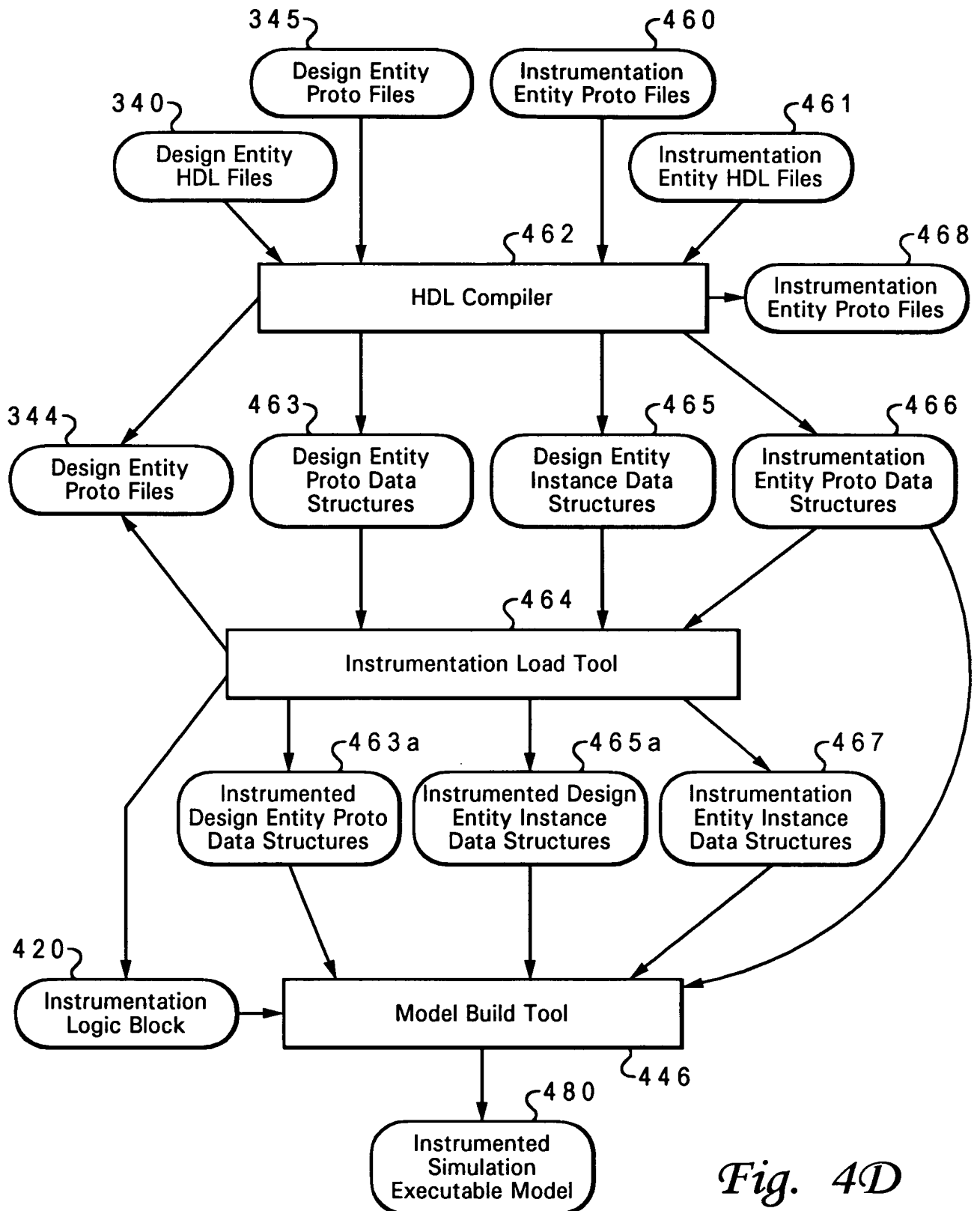


Fig. 4D

11/22

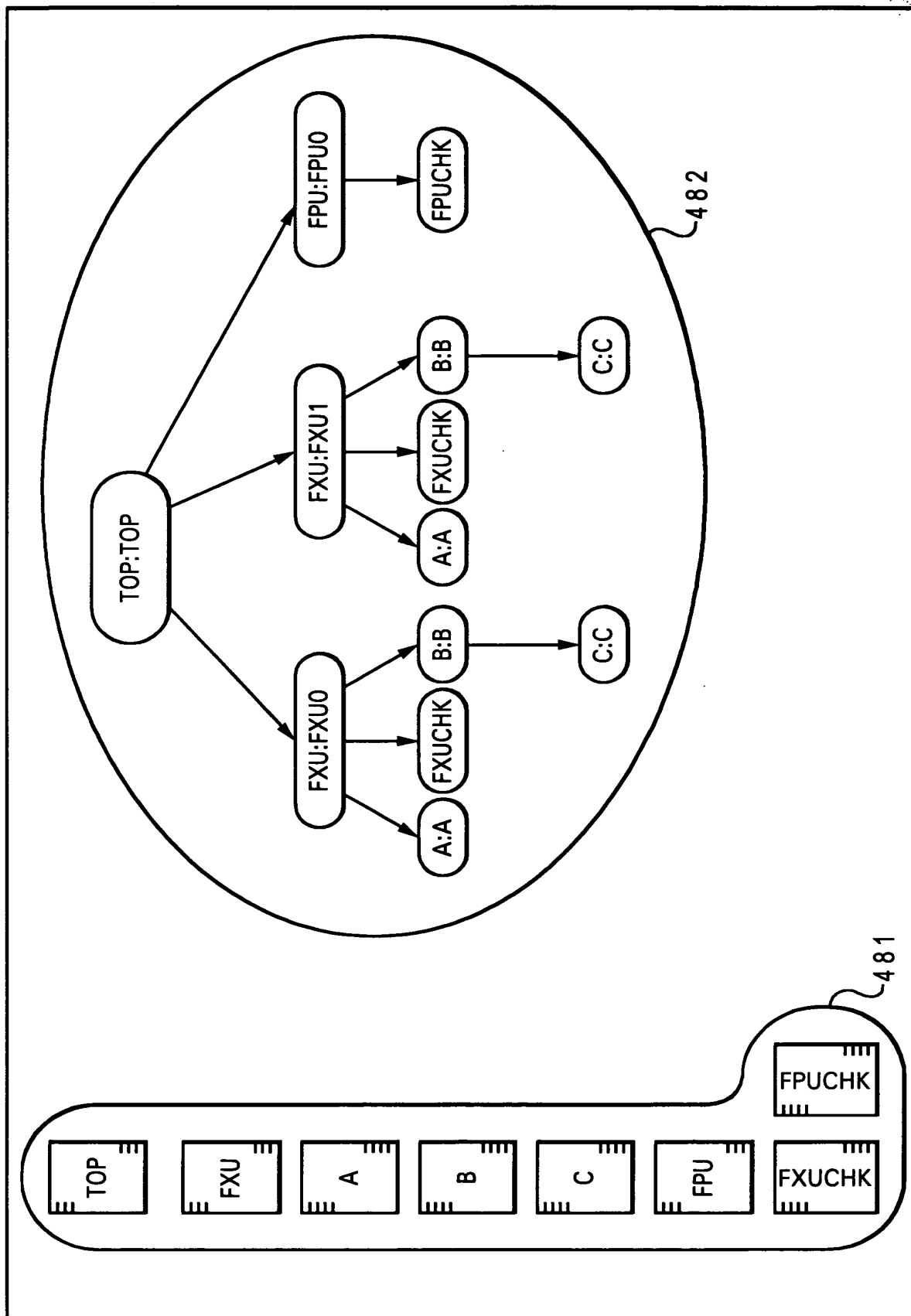


Fig. 4E



12/22

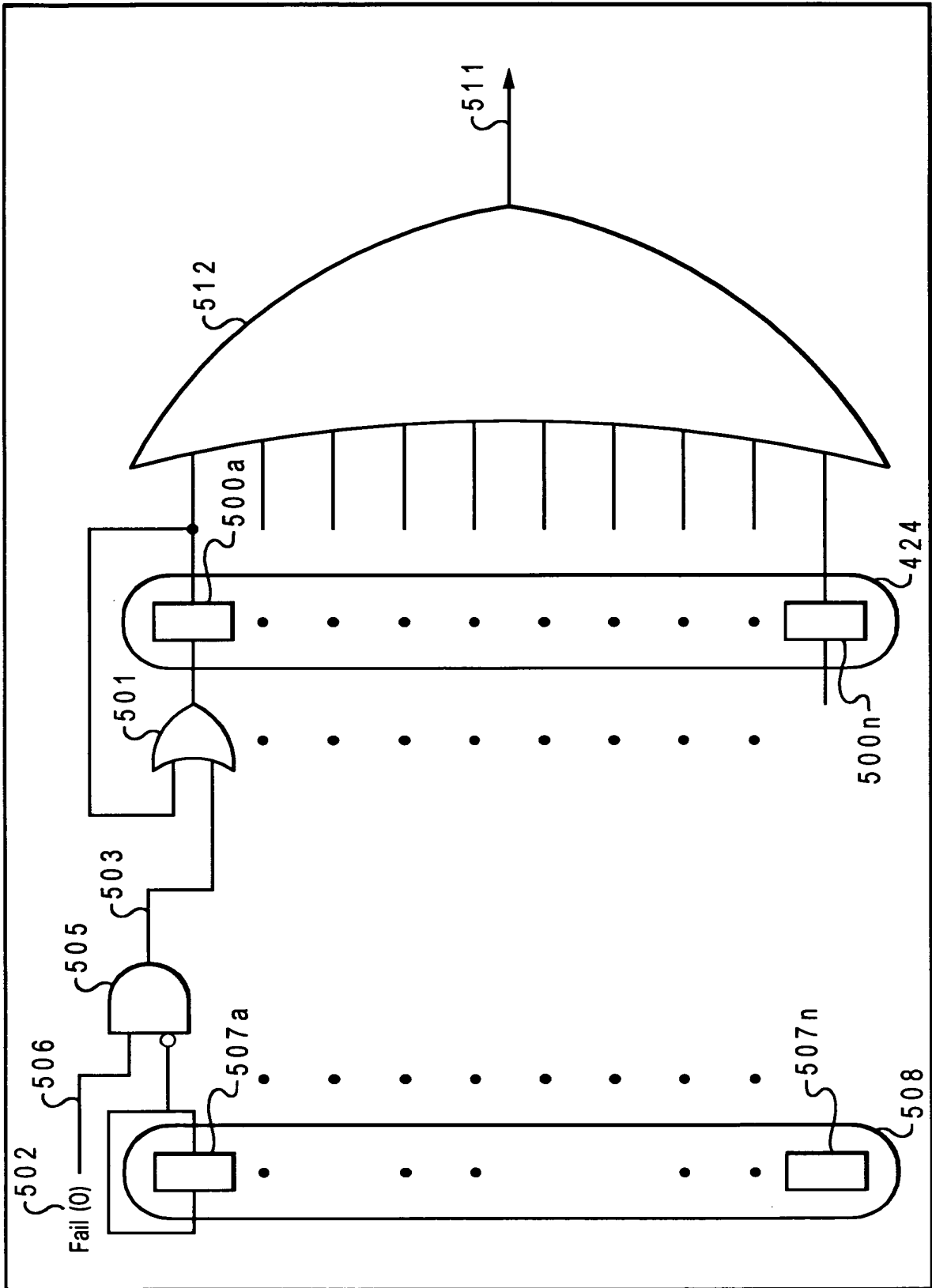
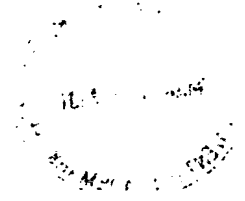


Fig. 5A

09751802



AUS920000224US1

13/22

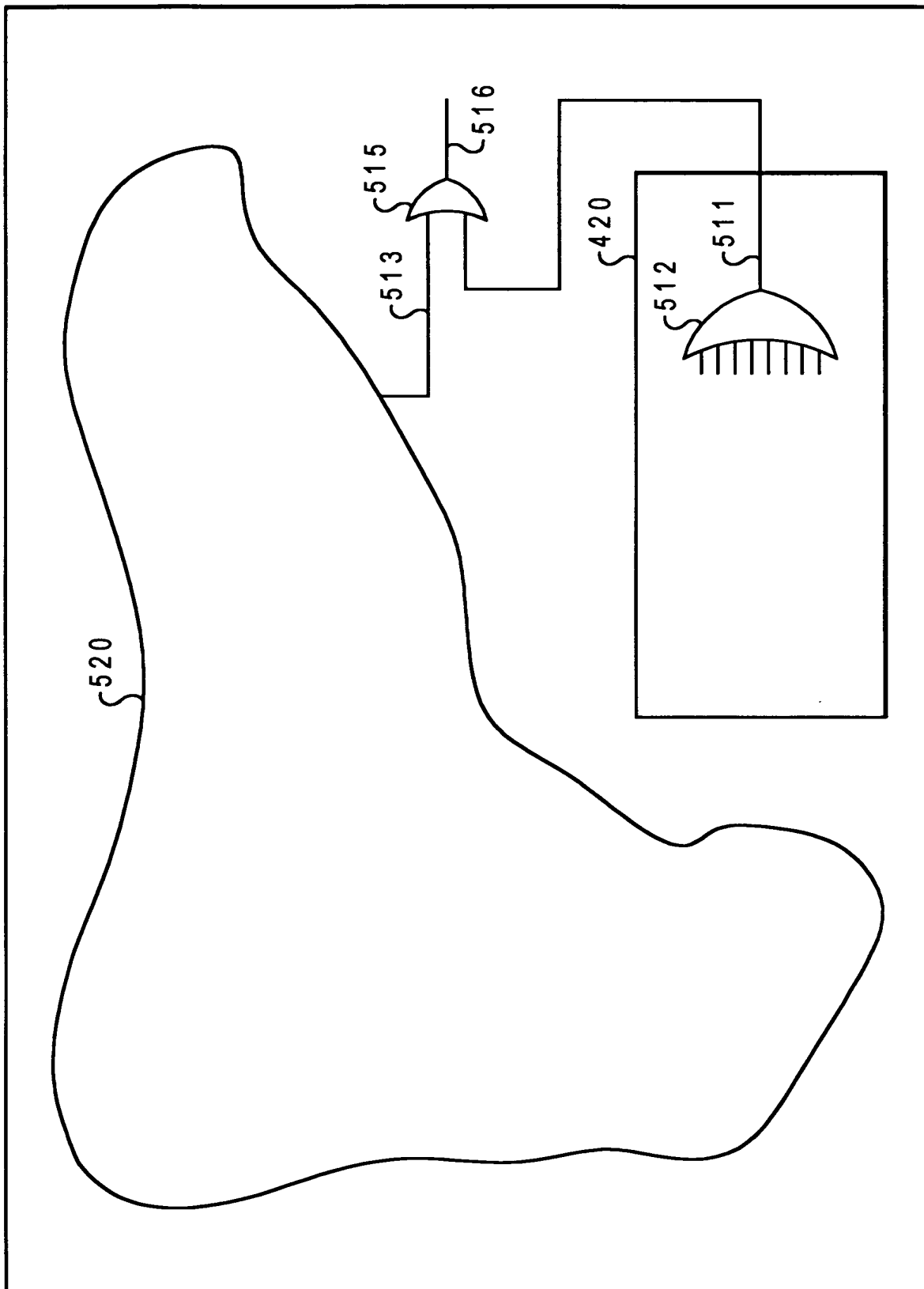


Fig. 5B

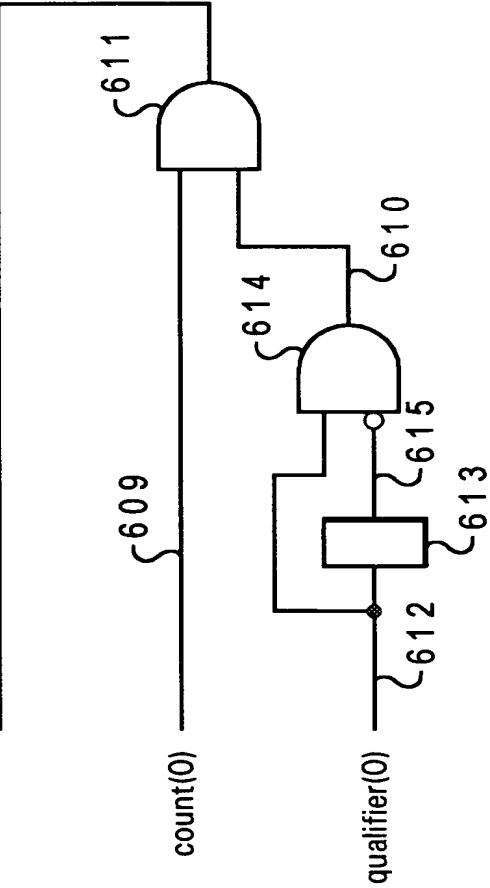
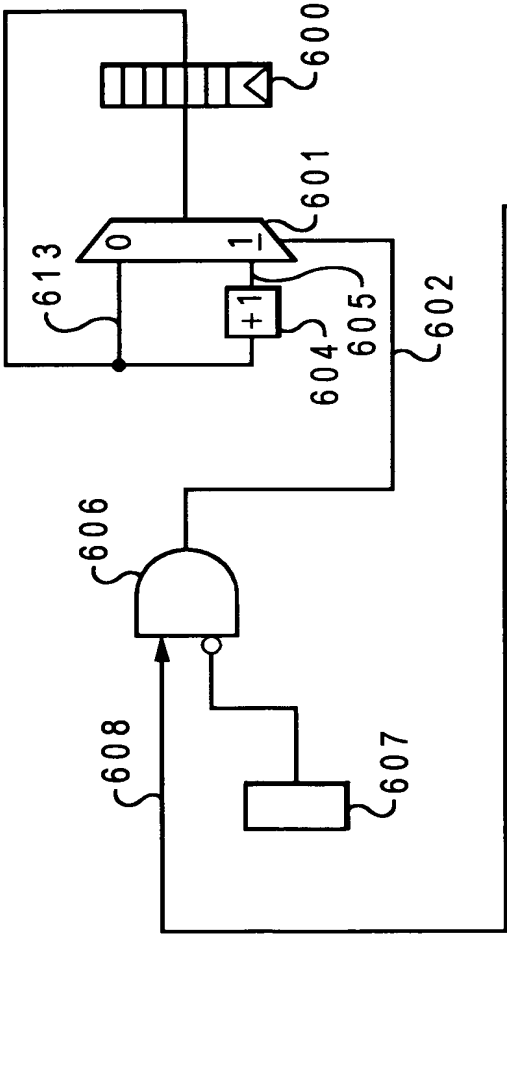


Fig. 6A

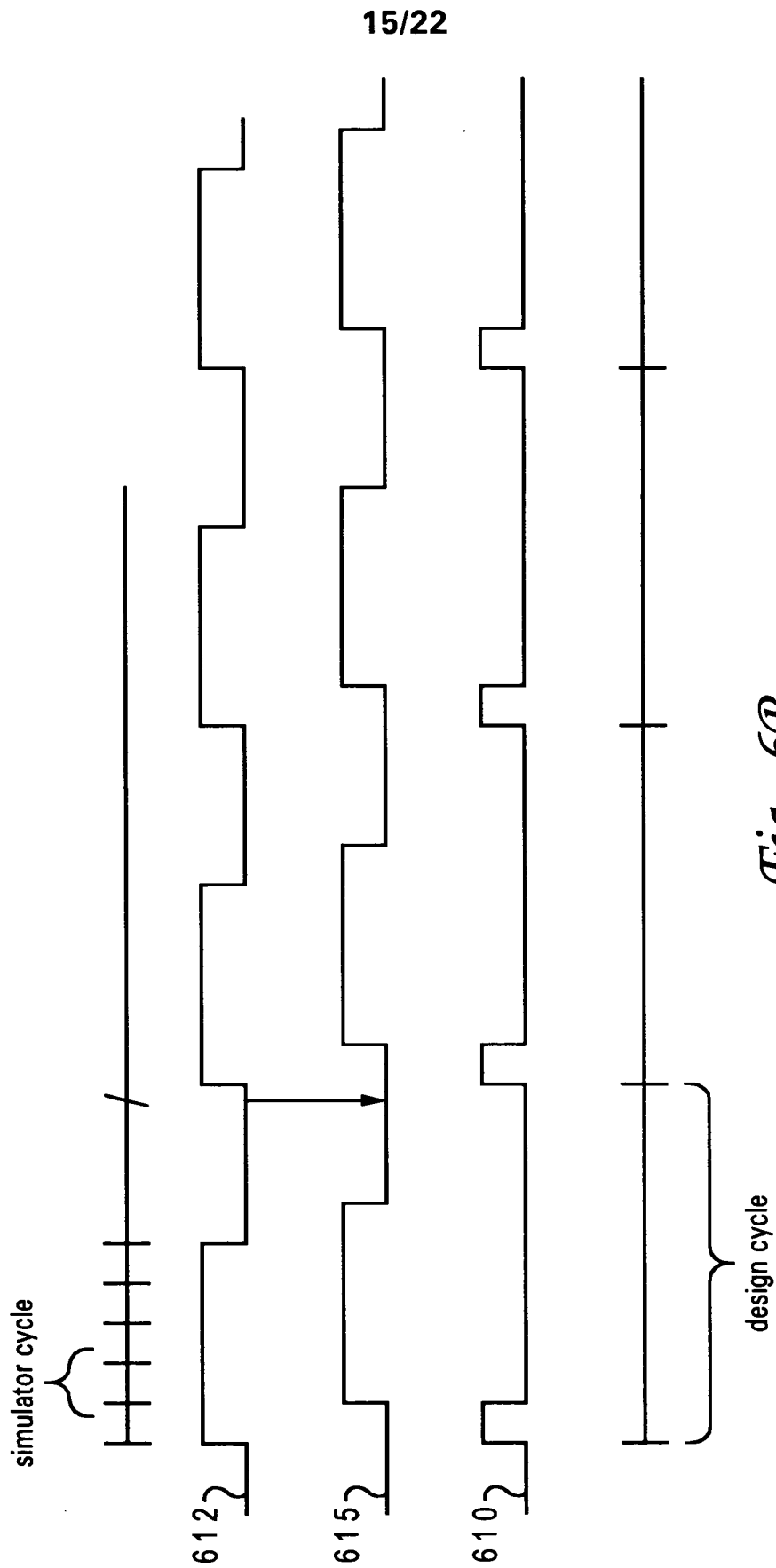
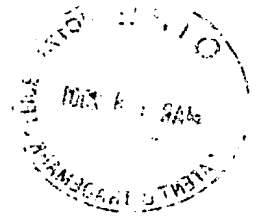


Fig. 6B

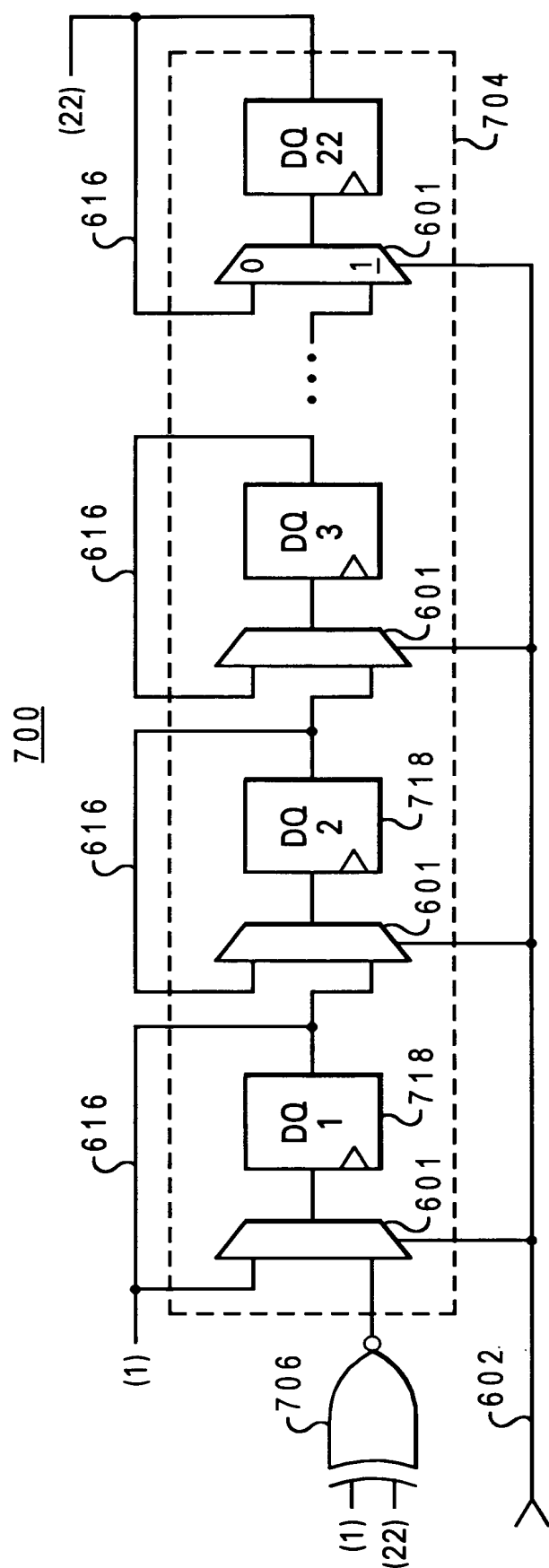
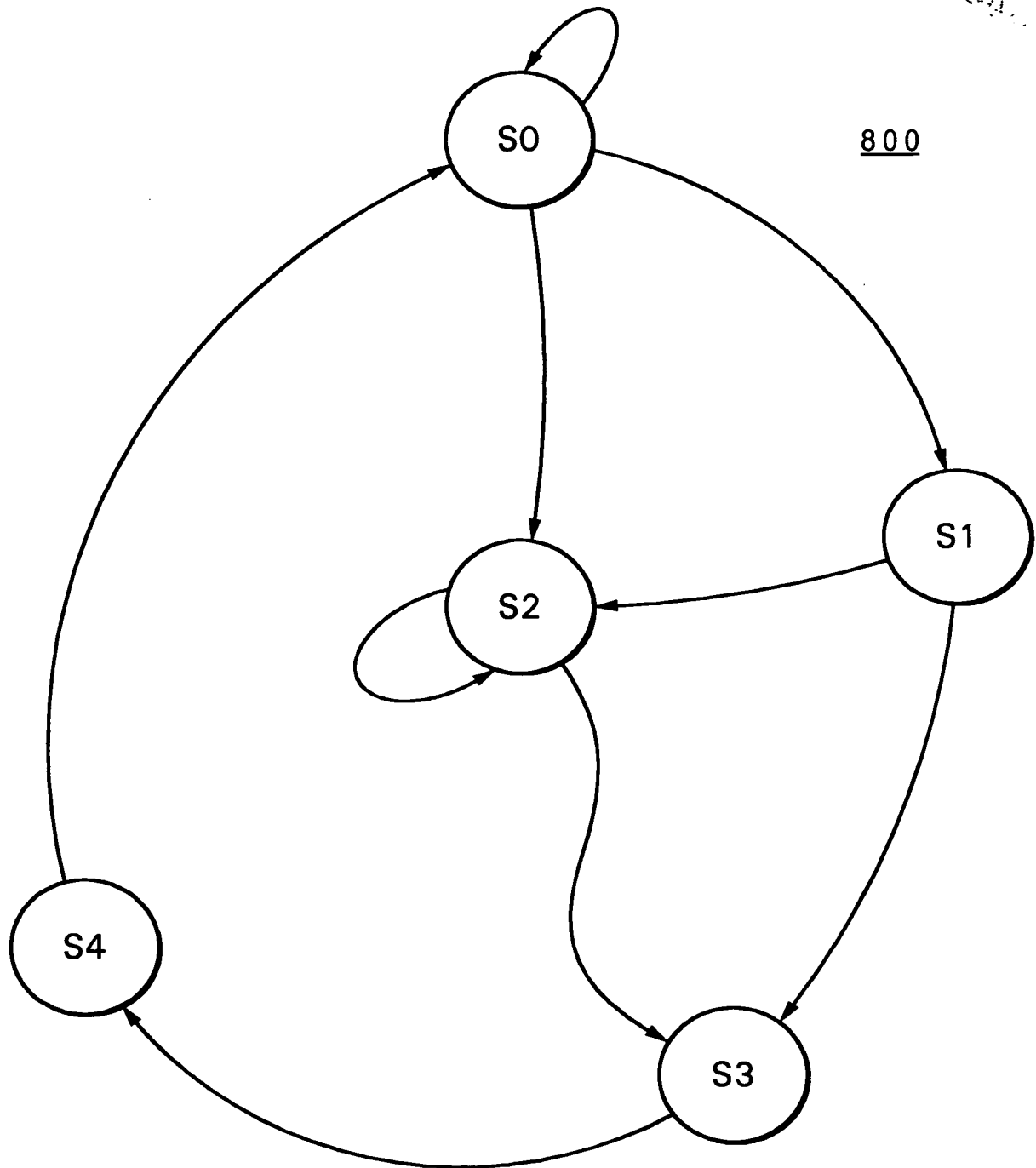
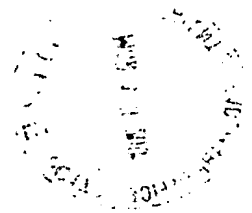


Fig. 7

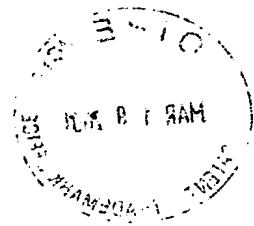


17/22



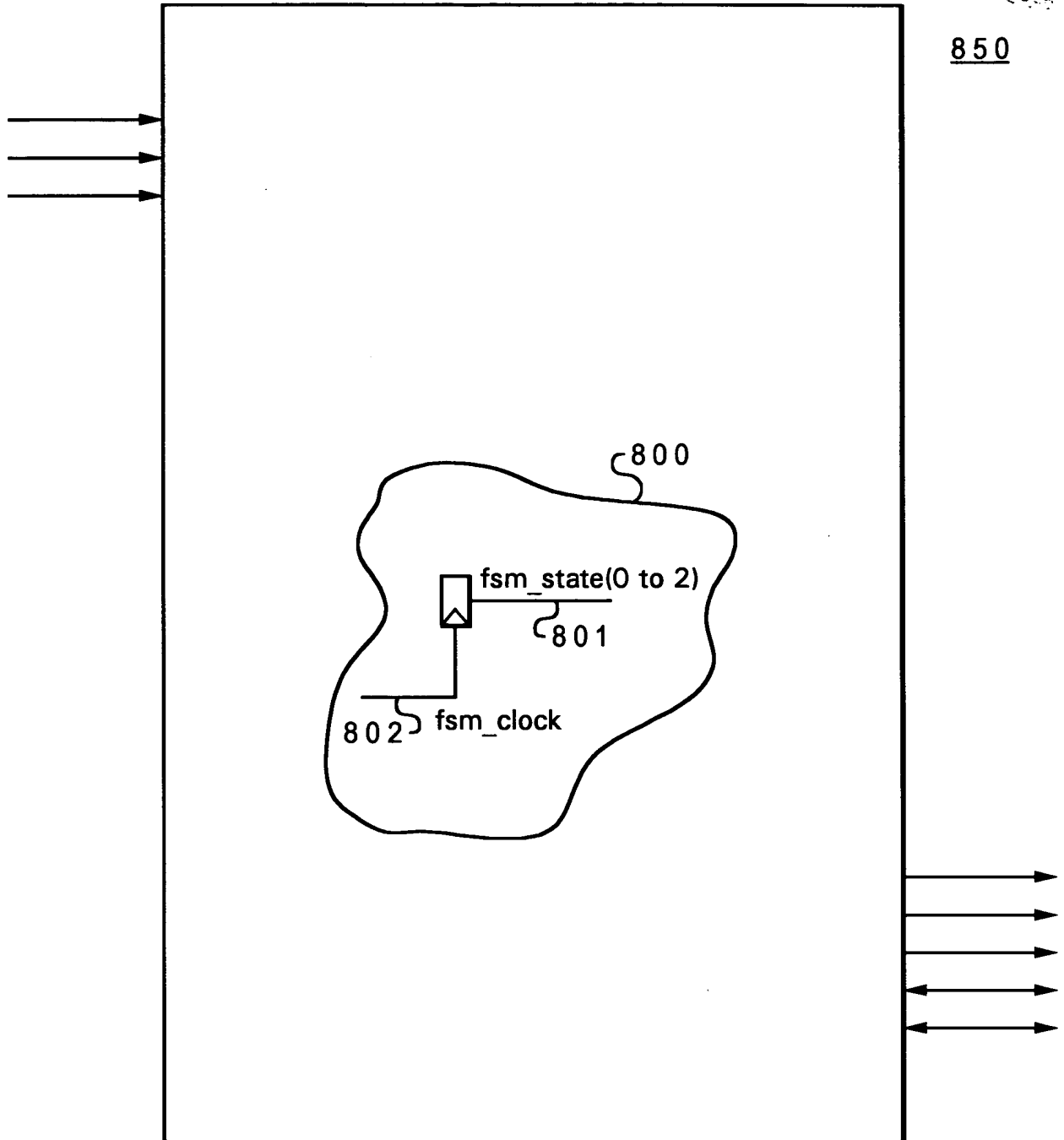
*Fig. 8A*  
*Prior Art*

18/22



entity FSM : FSM

850



*Fig. 8B*  
*Prior Art*

19/22

ENTITY FSM IS

```

PORT(
    ....ports for entity fsm....
);

```

ARCHITECTURE FSM OF FSM IS

BEGIN

... HDL code for FSM and rest of the entity ...

fsm\_state(0 to 2) &lt;= ... Signal 801 ...

```

8 5 3 { --!! Embedded FSM : examplefsm;
8 5 9 { --!! clock          : (fsm_clock);
8 5 4 { --!! state_vector   : (fsm_state(0 to 2));
8 5 5 { --!! states         : (S0, S1, S2, S3, S4);
8 5 6 { --!! state_encoding : ('000', '001', '010', '011', '100');
      { --!! arcs          : (S0 => S0, S0 => S1, S0 => S2,
8 5 7 { --!!                (S1 => S2, S1 => S3, S2 => S2,
      { --!!                (S2 => S3, S3 => S4, S4 => S0);
8 5 8 { --!! End FSM;

```

8 5 2 } 8 6 0

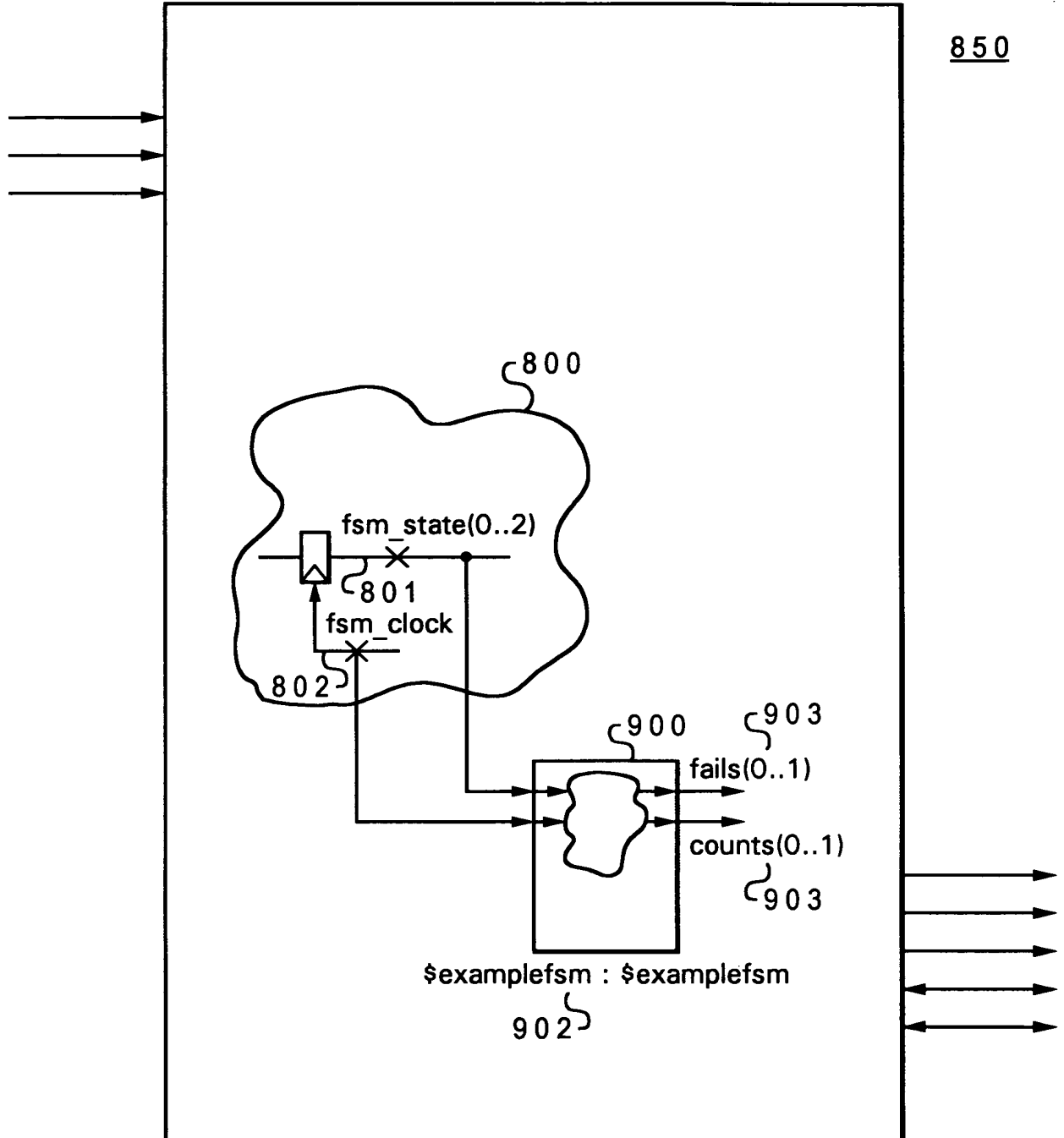
END;

*Fig. 8C*

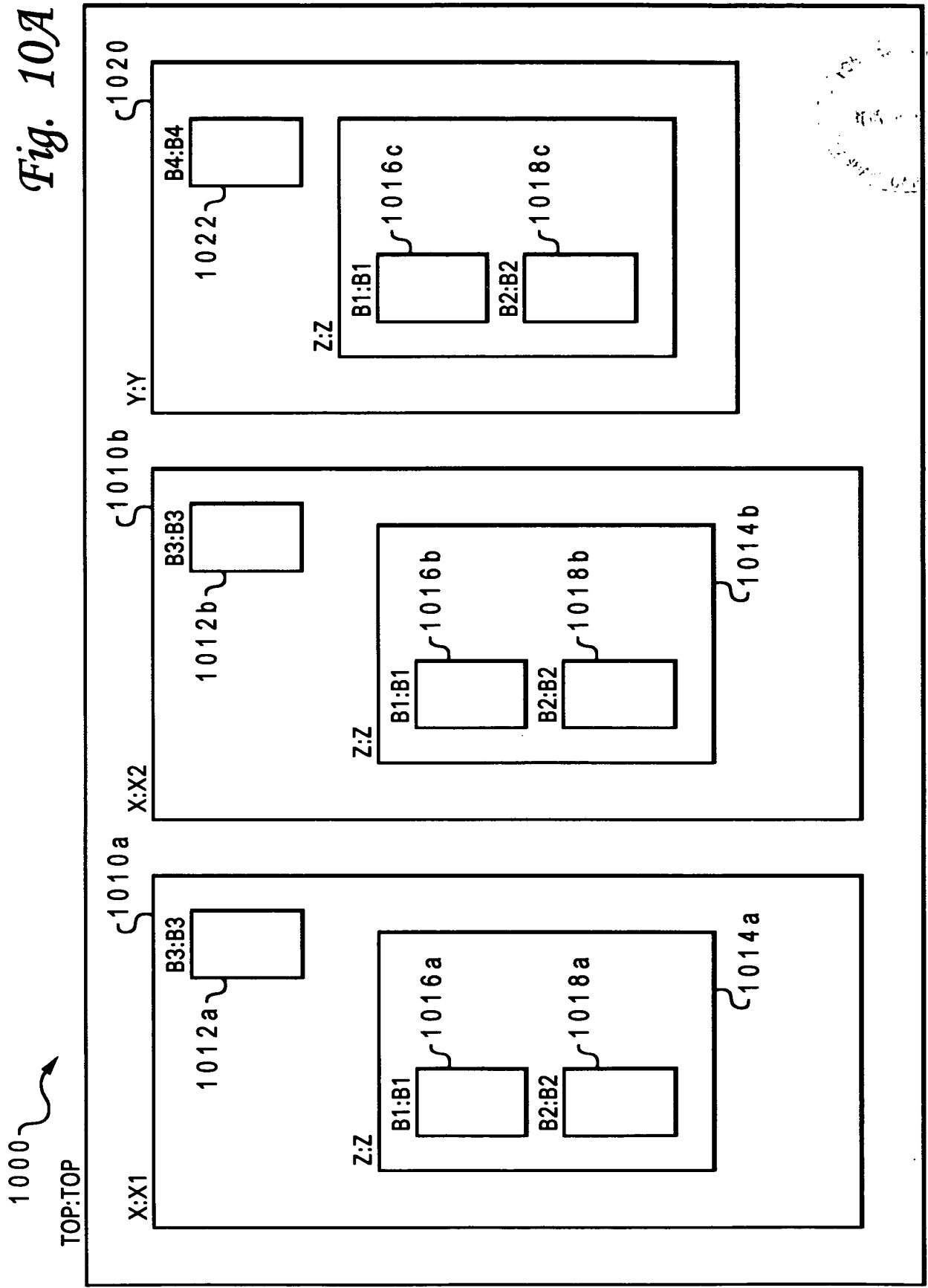


entity FSM : FSM

850



*Fig. 9*



1030 1032 1034 1036

<instantiation identifier>. <instrumentation entity name>. <design entity name>. <eventname>

*Fig. 10B*

1030	1032	1034	1036
↙	↙	↙	↙
X1	B3	COUNT1	1040
X1.Z	B1	COUNT1	1041
X1.Z	B2	COUNT1	1042
X2	B3	COUNT1	1043
X2.Z	B1	COUNT1	1044
X2.Z	B2	COUNT1	1045
Y	B4	COUNT1	1046
Y.Z	B1	COUNT1	1047
Y.Z	B2	COUNT1	1048

*Fig. 10C*

1030 1034 1036

<instantiation identifier>.<design entity name>.<eventname>

*Fig. 10D*